Comparación y evaluación de modelos: revisión de otros modelos y algoritmos

Bibliografía:

- An Introduction to Statistical Learning with Applications in R. 2017, Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani
- Giudici. Data Mining Model Comparison. Cap 32 de DM&KD Handbook. Maimon-Rokach Editors. Springer.
- Giudici&Figini. Applied Data mining for business and industry. Cap 5
- Albert, J. Bayesian Computation with R. (2009). Springer.

Modelos o algoritmos

- Para regresión:
  - Ridge
  - Lasso
  - ElasticNet
  - …

- Para clasificación:
  - Redes Bayesianas
  - Support vector machines

Regresión Ridge y Regresión Lasso

Regresión Ridge

- Se propone ante situaciones de multicolinealidad entre las regresoras.
- Propone sumar una pequeña cantidad positiva a cada elemento de la diagonal en el sistema [X'X + K] β = X'Y
- La estimación de los coeficientes bajo el método

$$\hat{\beta}ridge = argmin\,\beta \left\{ \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right\}$$

donde $\lambda \geq 0$ es un parámetro de complejidad que controla la magnitud de la penalización.

Regresión Ridge

- Reescribimos la ecuación anterior para hacer explícita la restricción sobre los parámetros

$$\hat{\beta}^{ridge} = argmin\,\beta \sum_{i=1}^{N}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 \qquad \sum_{j=1}^{p}\beta_j^2 \leq t$$

- El intercepto β_0 no se encuentra penalizado. $\widehat{\beta_0} = \frac{1}{N}\sum_{1}^{N} y_i$

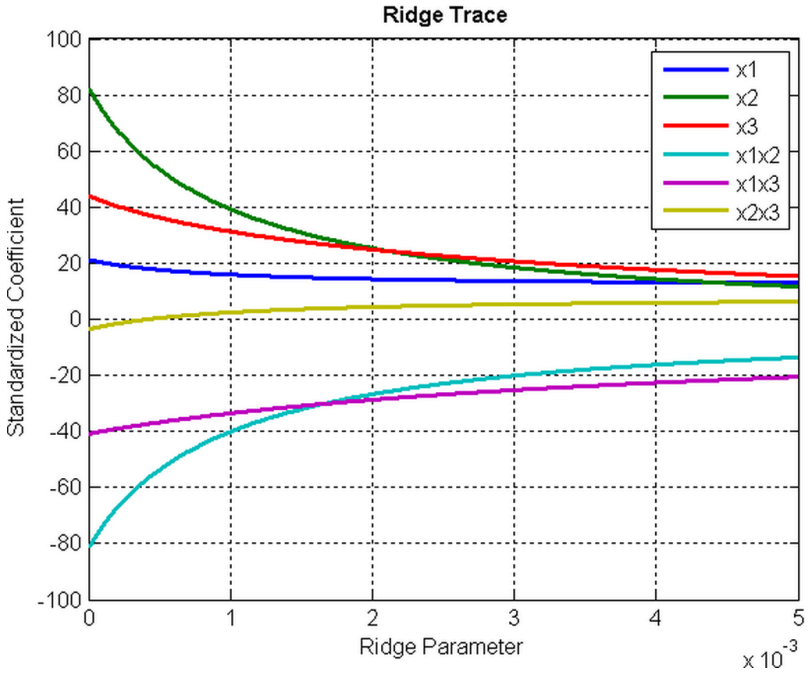
- Solución $$\hat{\beta}^{ridge} = (X'X + \lambda I)^{-1} X'Y$$

Cómo seleccionar λ?

La traza Ridge es un método clásico..

Lo que se usa habitualmente es Cross Validation!! Se recorre valores posibles de λ, y se evalúa CV-error para cada uno (por ejemplo, MSE).

Se elige λ para minimizar este.



Regresión Lasso

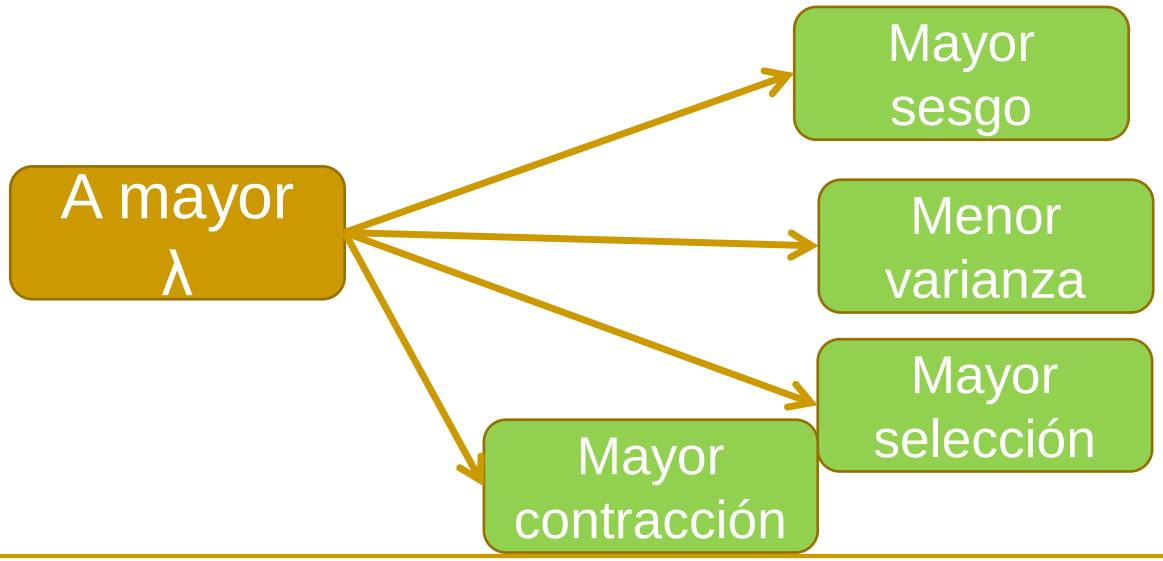
- Lasso → Least Absolute Shrinkage and Selection Operator (Operador de selección y contracción).
- Esta técnica encoge el modelo si existen variables que no aportan información o poseen problemas de colinealidad.
- Penaliza el método de mínimos cuadrados con una penalidad de tipo L1.

Formulación del Modelo

- La restricción propuesta es:
  obteniendo la siguiente ecuación para el problema:

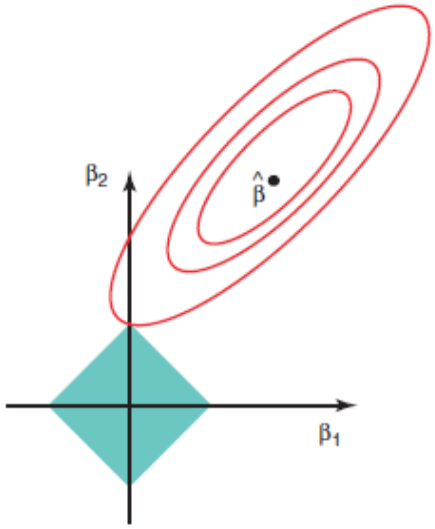
$$\sum_{j=1}^{p} \left|\hat{\beta}^{L}{}_{j}\right| \leq t$$

$$\hat{\beta}^{L} = \arg\min\{\sum_{t=1}^{N}\left(y_i - \alpha - \sum_{j}\beta_j - x_{ij}\right)^2 + \lambda * \sum_{j}|\beta_j|\}$$

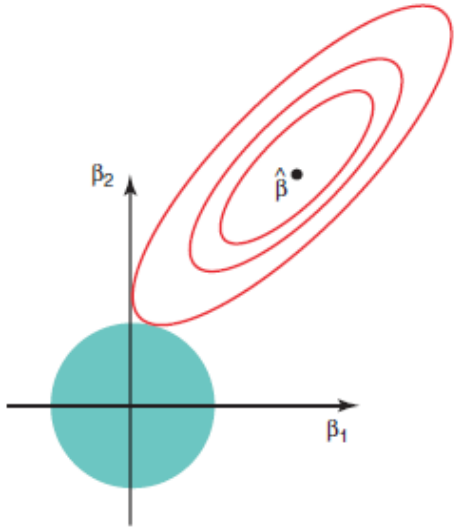


Regresión Lasso: Representación Gráfica

- $\hat{\beta}^0$ → estimaciones por mínimos cuadrados
- Área dentro del cuadrado → satisfacción de la condición de la regresión Lasso



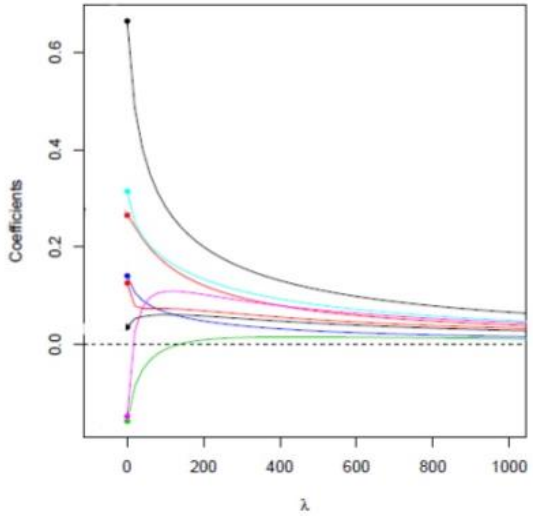
Regresión Lasso



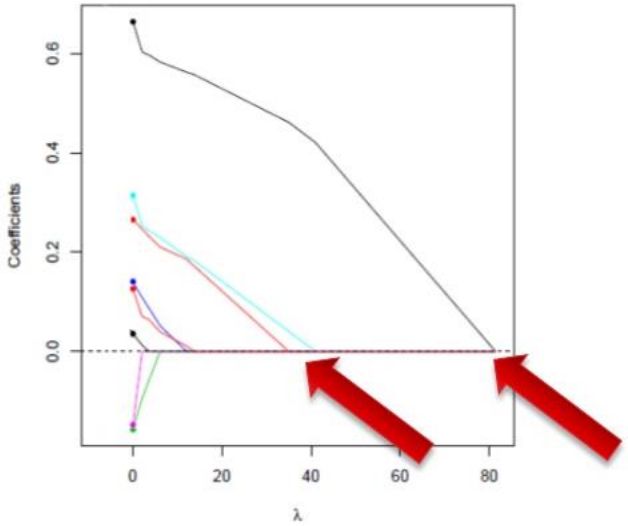
Regresión Ridge

Diferencias con RLM (estimación MCO)

- MCO no pone restricciones a los coeficientes, Ridge y Lasso si.
- Ridge posee una penalidad de tipo L2, y Lasso de tipo L1. Ridge no puede realizar selección de variables ya que nunca puede hacer que los coeficientes sean exactamente cero.



Regresión Ridge



Regresión Lasso

Regresión Lasso: Ventajas

- Poseen un buen error cuadrático medio (comparado con la regresión Ridge). Por este motivo, el método Lasso es muy competitivo con Ridge en cuanto a la reducción del error predicho.

- Debido a la posibilidad de seleccionar variables, este método obtiene modelos más simples y de interpretación más sencilla.

Regresión Lasso: Desventajas

- En los casos en que el número de parámetros es mayor al número de observaciones, el método Lasso no es el óptimo debido a que elegirá a lo sumo *n* variables de entre las candidatas. La regresión Ridge es un mejor estimador.

- En los casos de un grupo de variables altamente correlacionadas, la regresión Lasso tiende a seleccionar arbitrariamente una sola variable de dicho grupo. Esto presenta un problema, sobre todo en temáticas como pueden ser la selección de genes.

Conclusiones

Los tres métodos mencionados pertenecen a una misma familia. La elección de uno de ellos en particular dependerá de las características del problema a analizar y de los datos.

- ***Ridge*** → cuando no haya una solución única para el estimador de mínimos cuadrados, en la presencia de multicolinealidad severa.
- ***Lasso*** → para aquello problemas con sparse features (características dispersas).

Ejemplo: Comparación de regresiones

- data Hitters (library ISLR)
- Regresión múltiple versus ridge y lasso

Redes Bayesianas

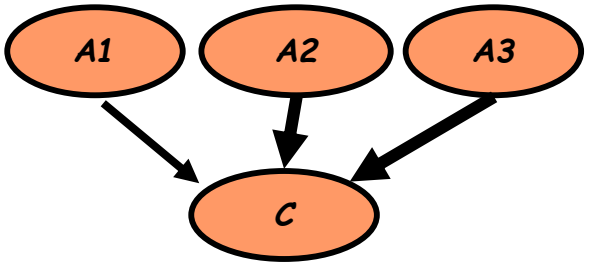
Redes Bayesianas (BN)

- Son estructuras gráficas-simbólicas para representar relaciones probabilísticas entre variables.
- Permiten definir modelos para diagnóstico (hallar P de la causa dados los síntomas) ó predicción (estando presente la causa, hallar P de los síntomas).
- Los nodos pueden se fuente de información u objeto de predicción, según sea la evidencia disponible.
- Se representan con un grafo DAG donde los los nodos representan variables aleatorias y los arcos representan dependencias entre ellas.

Modelos gráficos probabilísticos

Grafo DAG (*Directed Acyclic Graph*): cuando no existen caminos dirigidos hacia un mismo nodo (o sea no puede volverse al mismo nodo).

Ejemplo: 4 variables dicotómicas relacionadas por el sig grafo:



Para dar la distribución conjunta P(a1,a2,a3,c) necesitamos conocer

2^4 – 1=15 parámetros.

Si se tiene que A1,A2,A3 son independientes y C depende de todos los restantes, la conjunta es

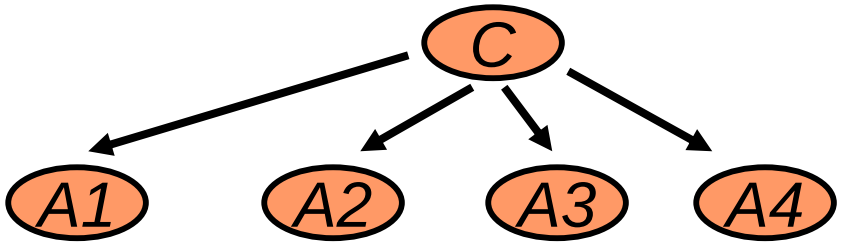
$$P(A1)P(A2)P(A3)P(C=c/A1,A2,A3)$$

sólo necesitamos conocer 11 parámetros!

Esto da la idea de que conocer estructuras de dependencia/indep nos permite reducir la información necesaria.

Otro ejemplo:

Si la estructura fuera:



Para dar la distribución conjunta P(a_1,a_2..,a_4,c) necesitamos conocer

2^5 – 1 =31 parámetros.

Si los arcos indican relaciones de dependencia, sólo necesitamos conocer

P(A1=0 /C=0); P(A1=0 /C=1);.. P(A4=0 /C=0); P(A4=0 /C=1); P(C=1)

que hacen un total de 4*2+1 =9 parámetros.

Definición

Una **Red Bayesiana (BN)** es un modelo gráfico probabilístico dado por

- un grafo DAG donde los nodos representan las variables y
- una distribución (conjunta) de probabilidades para las variables

tales que se verifica la **Condición de Markov:**

Cada nodo es independiente de sus no descendientes (nd) dado el conjunto de sus padres. Esto es:

$$X \perp nd(X) \mid pa(X)$$

Equivalentemente:

$$X \perp \{nd(X) - pa(X)\} \mid pa(X)$$

Cálculo de la distribución conjunta usando Markov

Teorema de Markov

Si un DAG con una función de probabilidad conjunta satisface la condición de Markov (esto es, se tiene una BN) entonces vale:

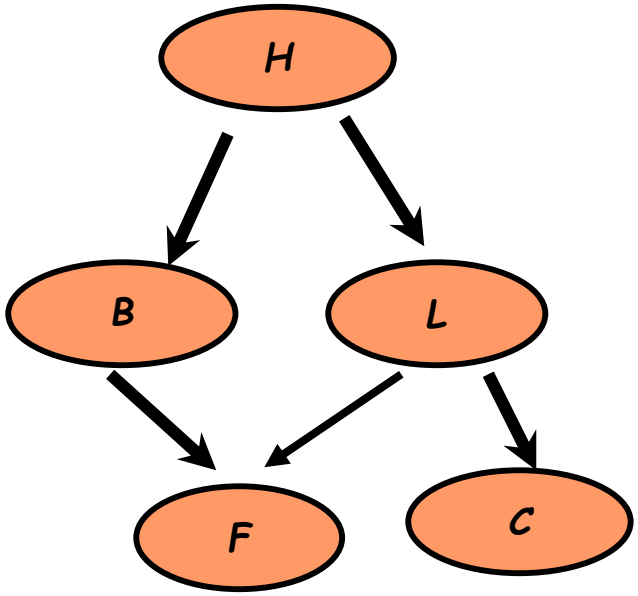
$$P(x_1, x_{2,}..x_n) = \prod_{i=1}^{n} P(x_i \,/\, pa(x_i))$$

Donde para los nodos raíz, $P(X/pa(X)) = P(X)$

Obs: para variables discretas o Normales vale también la recíproca: si la conjunta se factoriza según esta expresión, entonces se cumple la condición de Markov.

Ejemplo de BN (Neapolitan, pag 4):

Se propone la siguiente estructura de BN para las variables:



H: historia de fumador?

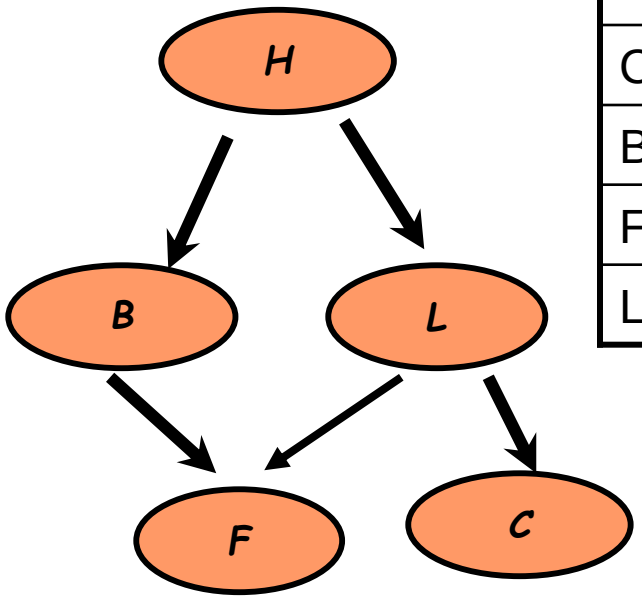
B: tiene bronquitis?

L: tiene cáncer de pulmón?

F: tiene fatiga pulmonar?

C: RX de torax positivo?

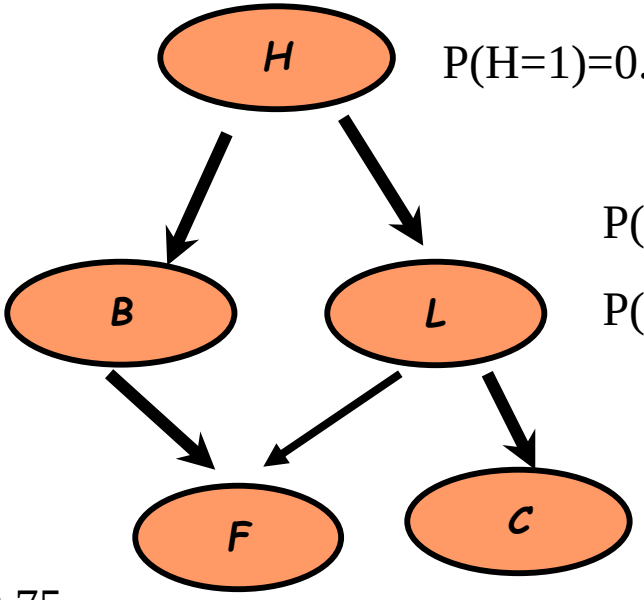
Ejemplo de BN (Neapolitan, pag 4):



Nodo	Pa	nd	indep
C	L	H,B,F,L	C ⊥ {H,B,F} |L
B	H	H,L,C	B ⊥ {L,C} |H
F	B,L	H,B,L,C	F ⊥ {H,C} | {B,L}
L	H	H,B	L ⊥ {B} |H

$$P(H,B,L,F,C) = P(H)P(B/H)P(L \mid H)P(C \mid L)P(F \mid B,L)$$

Si conocemos los parámetros (la distribución de las variables) y la estructura, tenemos la distribución conjunta



P(H=1)=0.2

P(B=1/H=1)=0.25

P(B=1/H=0)=0.05

P(L=1/H=1)=0.03

P(L=1/H=0)=0.00005

P(C=1/L=1)=0.6

P(C=1/L=0)=0.02

P(F=1/B=1,L=1)=0.75

P(F=1/B=1,L=0)=0.10

P(F=1/B=0,L=1)=0. 5

P(F=1/B=0,L=0)=0.05

Para dar la distribución necesitamos 11 parámetros

$$P(H,B,L,F,C) = P(H)P(B/H)P(L|H)P(C|L)P(F|B,L)$$

Cuestiones a resolver:

I. Fijar la estructura de la red (aprendizaje estructural)

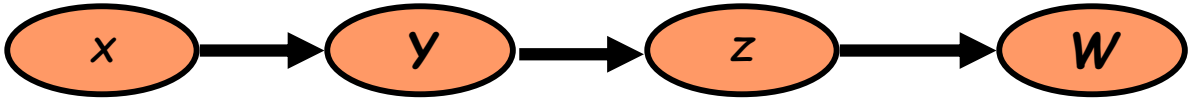
II. Determinar la distribución conjunta de las variables involucradas (aprendizaje paramétrico)

III. Una vez determinada la red, hacer inferencia para alguna variable conocidas las demás (propagación de la evidencia).

Inferencia: propagación de probabilidades

¿Cómo deducimos las probabilidades de unas variables conocidos los valores observados de otras?

Ejemplo (simplísimo): dada la siguiente red bayesiana



P(X1)=0.4

P(Y1/X1)=0.9

P(Y1/X2)=0.8

P(Z1/Y1)=0.7

P(Z1/Y2)=0.4

P(W1/Z1)=0.5

P(W1/Z2)=0.6

Si X=X1, cómo se propaga esta información? Esto es, cómo resulta por ejemploP(Z1/X1)?

Si W=W1, cómo se propaga esta información? Esto es, cómo resulta P(X1/W1)?

Aprendizaje paramétrico

Inicialmente la estructura era construída por expertos, como es el caso del ejemplo de cáncer de pulmón.

Una vez que la estructura está determinada, se quiere aprender la distribución conjunta a partir de los datos. El método más aceptado para estimar los parámetros de la distribución conjunta (i.e., aprendizaje paramétrico) es el de máxima verosimilitud o variantes de éste.

Esto corresponde, para variables discretas, a la estimación mediante frecuencias observadas en la muestra.

Para el caso de nodos ocultos ó datos faltantes se puede usar EM.

Aprendizaje paramétrico

Lo anterior muestra que la verosimilitud a maximizar es producto de verosimilitudes "separables", cada una de las cuales corresponde a una variable Binomial.
Luego el máximo corresponde a maximizar cada una de ellas, lo que convierte en un problema más sencillo.

Recordando que el estimador MV del parámetro θ en una muestra de variables Bernoulli (o Binomial) resulta en la frecuencia relativa:

$$\theta_{MV} = \frac{N_k}{N}$$

Aprendizaje paramétrico

Para una variable discreta con k clases se tiene la verosimilitud de una multinomial:

$$L(\Theta : D) = \prod_{k=1}^{K} \theta_k^{N_k}$$

Donde θk es la probabilidad de que se presente el caso k y Nk es la cantidad de veces que aparece en la muestra.

El estimador MV de θk es el resultado de buscar el máximo a esta función, lo que resulta en la frecuencia relativa: $\theta_k = \frac{N_k}{N}$

Aprendizaje paramétrico

Proponiendo una alternativa bayesiana se tienen las siguientes formas de estimar los parámetros para el caso de variables discretas:

$$\hat{\theta}_{x_i|pa_i} = \frac{N(x_i, pa_i)}{N(pa_i)}$$

MLE

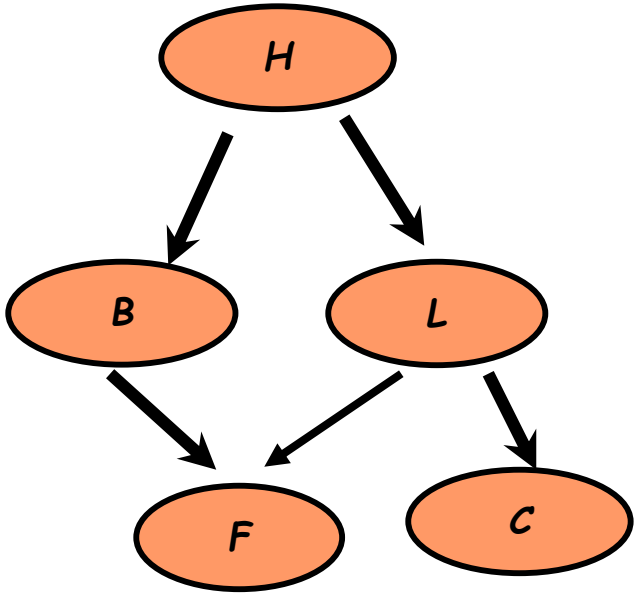
$$\tilde{\theta}_{x_i|pa_i} = \frac{\alpha(x_i, pa_i) + N(x_i, pa_i)}{\alpha(pa_i) + N(pa_i)}$$

Bayesiano (Dirichlet)

Ambas son asintóticamente equivalentes

Ejemplo

En este caso



$$P(h=1) = \frac{\#casos\ (h=1)}{\#total}$$

$$P(l=1/h=1) = \frac{\#casos\ (l=1 \wedge h=1)}{\#casos\ (h=1)}$$

$$P(f=1/b \wedge h) = \frac{\#casos\ (f=1 \wedge b \wedge h)}{\#casos\ (b \wedge h)}$$

etc...

Aprendizaje estructural

El primer paso para ajustar una RB es especificar su estructura. Esto puede hacerse basado en rutinas automáticas ó en juicio de expertos.

El aprendizaje automático consiste en inducir, a partir de los datos, una estructura. Esto es, determinar las relaciones de dependencia e independencia entre las variables.

Redes Bayesianas para clasificación

Son ampliamente usados por varias ventajas:

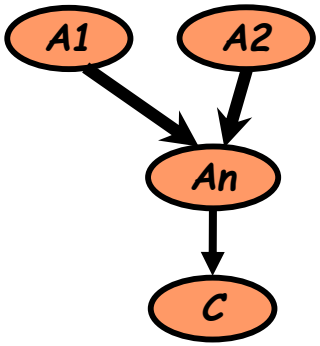
- fáciles de construír y entender
- la inferencia es sencilla
- son robustos a atributos irrelevantes

Clasificación con Redes Bayesianas

Supongamos un conjunto de variables C, A_1, A_2, …, A_n las cuales pueden pensarse como

C: variable de clasificación

A_i: atributos



¿Cómo predecir C a partir de los atributos?

- buscar el valor de C que maximice P(A_1, A_2,.., A_n/C) ➡ E. MV
- buscar el valor de C que maximice las probabilidades a posteriori P(C / A_1, A_2, …, A_n), o equivalentemente,

Maximizar P(A_1, A_2, …, A_n /C)P(C) ➡ E. Bayesiana

Clasificación con BN

La propuesta bayesiana implica calcular $P(C \mid A_1, A_2, \ldots, A_n)$ para todos los valores de C usando el teorema de Bayes

$$P(C \mid A_1 A_2 \ldots A_n) = \frac{P(A_1 A_2 \ldots A_n \mid C) P(C)}{P(A_1 A_2 \ldots A_n)}$$

Como el denominador no depende de C, sólo se necesitan para la clasificación definir las probabilidades a priori P(C) y luego conocer las probabilidades condicionales $P(A_i \mid C)$

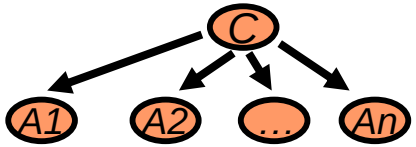
Clasificador **Naive Bayes**

Asume independencia entre los atributos, condicionados a C.

Entonces

$$P(A_1,..|C_i) = \prod_{k=1}^{n} P(a_k|C_i) = P(a_1|C_i)\times P(a_2|C_i)\times...\times P(a_n|C_i)$$

Con lo que

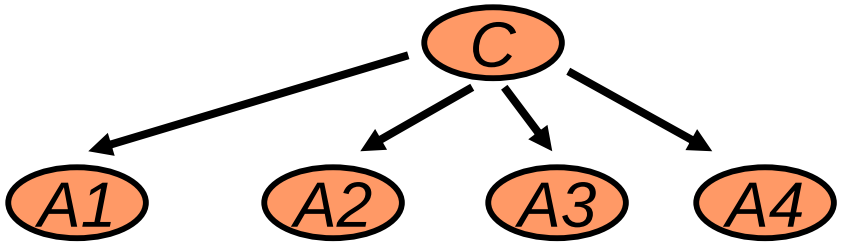


$$P(C_i|A_1A_2\ldots A_n) \propto \prod_{k=1}^{n} P(a_k|C_i)P(C_i)$$

La clasificación se hace maximizando esta última expresión.

Ejemplo de Naïve Bayes:

Para una variable clasificatoria con *n=4* atributos binarios:



Necesitamos conocer :

* $P(A_i = 0 / C=0)$ y $P(A_i = 0 / C=1)$ para cada i=1..4
(las verosimilitudes de los datos)
* P(C=0) *(la distribución a priori de C)*

Cómo estimarlas desde los datos?

- Para la distribución *a priori*:

$$P(C=c) = N_c/N$$

- Para la condicional de cada atributo:

$$P(A_i \mid C) = |A_i|/ N_c$$

  - donde $|A_{ic}|$ es el nº de casos A_i observados en la clase C
  - N_c es el nº de casos C observados

Naïve Bayes

Alternativas para estimar los parámetros:

Máxima verosimilitud: $P(A_i \mid C) = \frac{N_{ic}}{N_c}$

Laplace: $P(A_i \mid C) = \frac{N_{ic} + 1}{N_c + c}$

m-estimate: $P(A_i \mid C) = \frac{N_{ic} + mp}{N_c + m}$

Donde

Nc=cantidad de datos de la clase condicionante

c= # categorías de Ai (por ejemplo, c=2 para atributo binario)

m= tamaño equivalente de muestra (hay que fijarlo)

p: probabilidad a priori (=1/k si no hay información, k=nº val que puede tomar ese atributo)

Propiedades de Naïve Bayes

- Reduce significativamente el número de parámetros.
- Es robusto a puntos aislados.
- Es robusto a atributos irrelevantes.

Problemas:

- El supuesto de independencia puede ser muy fuerte en algunos casos.
- Para datos esparsos, la probabilidad a posteriori es nula (como en el ejemplo anterior), y no permite tomar en cuenta las probabilidades de los otros atributos observados, produciendo a veces mala clasificación.

Software

R:

naiveBayes (de *e1071 package*), bnlearn package, deal, gRain, …entre otros.

Weka:

Naive Bayes, Bayes Net, Simple Bayes.

Elvira, Hugin….etc

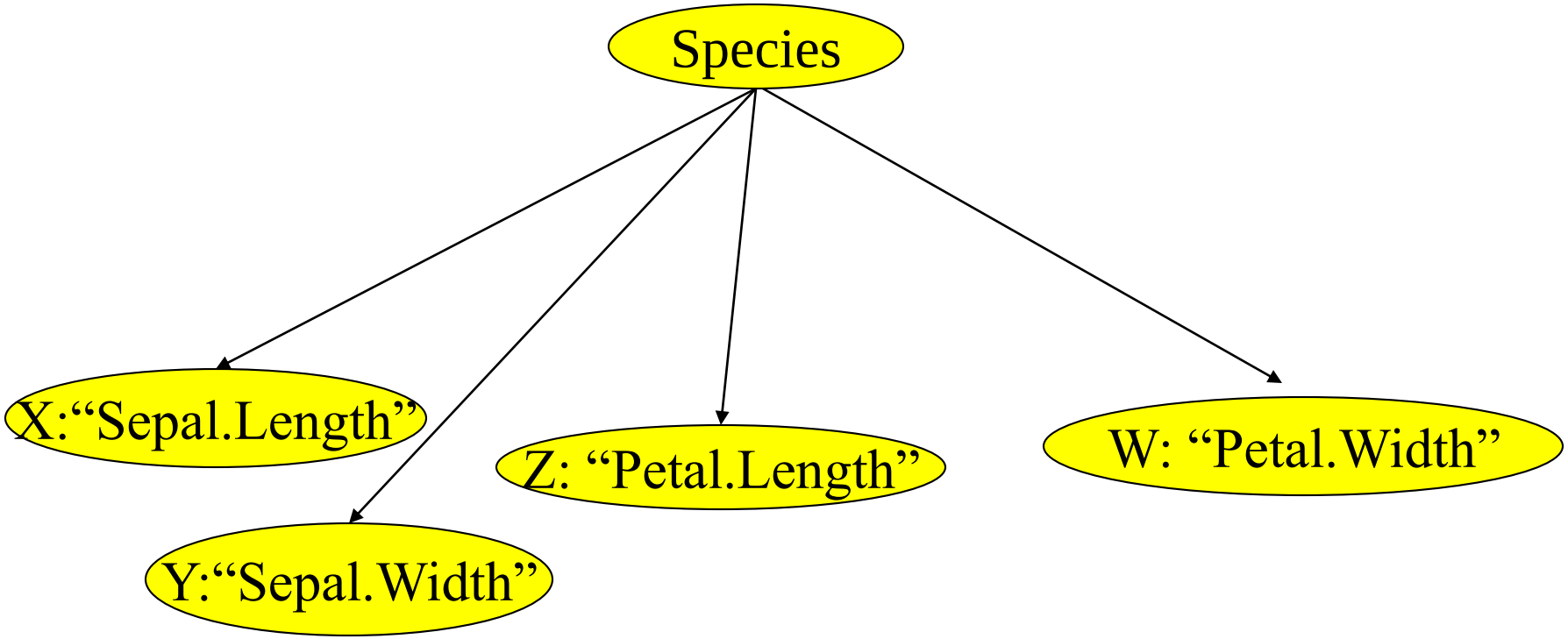
Ejemplo: Naive Bayes con datos iris

Los datos corresponden a 50 muestras para cada variedad de iris (Species: Iris setosa, Iris virginica y Iris versicolor) para las que se midió X:“Sepal.Length”, Y:“Sepal.Width”; Z: “Petal.Length” W: “Petal.Width”

usamos este paquete para construír un clasificador Naive Bayes:

```
install.packages("e1071")
clasificador<-naiveBayes(Species~.,iris)
```

Naive Bayes con datos iris



las variables predictoras son todas continuas, se suponen Normales.

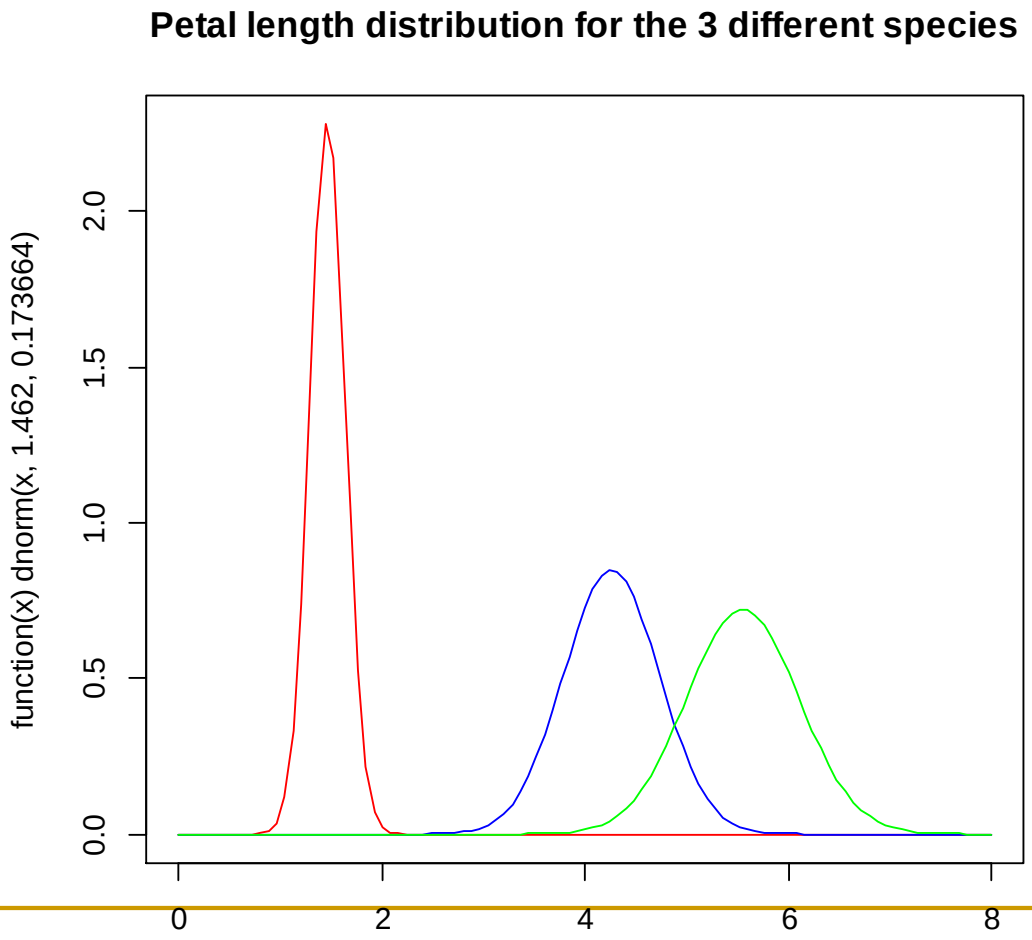
Lo siguiente da la media y la desviación de las 3 distribuciones dependientes (condicionales) para la variable Z:

```
clasificador$tables$Petal.Length
```

	Petal.Length	
species	[,1]	[,2]
setosa	1.462	0.1736640
versicolor	4.260	0.4699110
virginica	5.552	0.5518947

Ejemplo 1: iris

Graficamos las tres densidades normales ajustadas



Ejemplo: iris

Predicción: para clasificar una nueva observación ajustamos el modelo sin la última observación

```
irisT= iris[-150,]
clasificador1<-naiveBayes(irisT[,1:4], irisT[,5])
predict(clasificador1, iris[150,],type = "raw")
```

Y se tiene las probabilidades:

```
          setosa           versicolor       virginica
[1,]  3.000941e-143  0.06483664      0.9351634
```

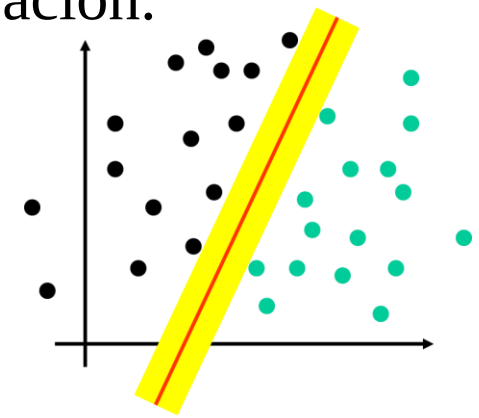
Lo que la clasificó como “virginica”. El dato original es:

```
iris[150,]
    Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
150          5.9           3          5.1         1.8 virginica
```

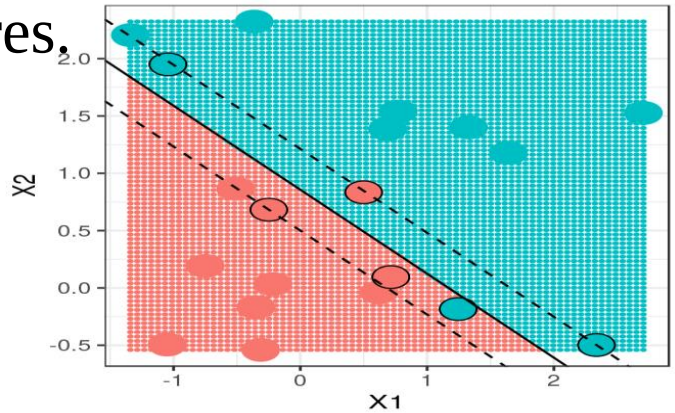
SVM

Máquinas de vector soporte (SVM)

Se busca un hiperplano que tenga el mayor margen de separación.



Se extiende el concepto para obtener un hiperplano que ¨casi¨ separe las clases, pero permitiendo que cometa unos pocos errores.



Cuando los datos no se pueden ¨separar¨ linealmente -> transformar los datos (aumenta dim) de manera que se puedan separar (***Kernel).***

Otra: separar con **funciones polinómicas, radial,etc.**

SVM clasificador

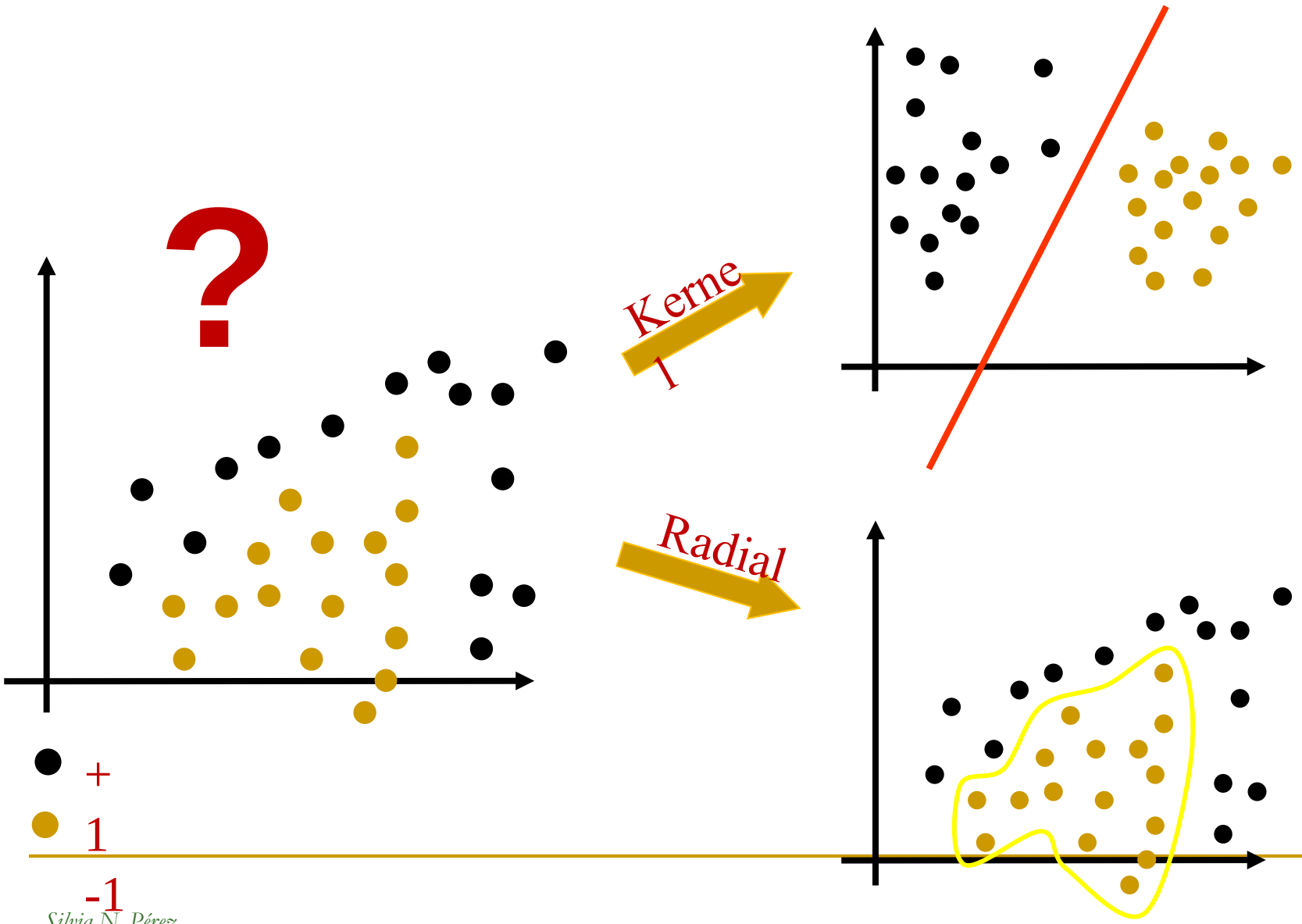
$$\underset{\beta_0,\beta_1,\ldots,\beta_p,\epsilon_1,\ldots,\epsilon_n}{\text{maximize}} \quad M$$

$$\text{subject to} \quad \sum_{j=1}^{p} \beta_j^2 = 1,$$

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip}) \geq M(1 - \epsilon_i),$$

$$\epsilon_i \geq 0, \quad \sum_{i=1}^{n} \epsilon_i \leq C,$$

SVM



Ejemplo: comparación en clasificación

- Data PrestamoSelect
- Comparación de Regresión logística con Naive Bayes y con SVM

Ejemplo Clasificación en datos de EP

- Dataset: Registros de individuos con enfermedad de Parkinson (https://www.synapse.org/).

- A los enfermos se ha grabado un fonema resultando en 38 variables medidas.
- Además se registró sexo, edad, entre otras variables.

- La clasificación se hace a través de los grados de severidad medidos según la suma de la escala PDRS-Parkinson'sDisease Rating Scale).
- La respuesta PDRS es autoreportada.

UPDRS - Unified Parkinson's Disease Rating Scale

- Escala Unificada de Valoración de la Enfermedad de Parkinson (UPDRS).
- Esta escala tiene valoraciones múltiples que miden el funcionamiento mental, la conducta, el ánimo; las actividades de la vida cotidiana y la función motora.
- El rango de la escala UPDRS varia desde "0"(no incapacidad) hasta incapacidad total.
- Los resultados son autoreportados por los pacientes.

Parte I: Estado mental, de comportamiento y humor.

Parte II: Actividades de la vida diaria.

Parte III: Examen motor.

Nuevo estudio de clasificación binaria en datos Synapse

- Se seleccionaron 731 casos filtrando aquellos que presentaron valores anormales.
- Se utilizó PDRS según valores bajos (pdrs<25) o altos. Variable de clase: pdrs2C
- Los modelos consideran:

pdrs2C~f1+f4+f7+f10+f12+f15+f18+f21+f27+f32+f33+f34+f35+f38+sex

aplicando sobre estas stepwise.

Table 1. Disphonia measures grouped into families of features

Label	Group	Features
$f_1 \ldots f_5$	Pitch measures	F0 and its derivative, RMS power
$f_6 \ldots f_{19}$	MFCC	Median cepstral coefficients 0-12
$f_{20} \ldots f_{32}$	MCC derivatives	Time derivatives of MCC 0-12
$f_{33} \ldots f_{35}$	Non linear measures	RPDE, DFA, PPE
$f_{36} \ldots f_{38}$	Relative spectral power	0-500Hz, 500Hz-1kHz, 1-2kHz

Nuevo estudio de clasificación binaria en datos Synapse

- Se partió el dataset en test/train (80/20) resultando 147 casos en testing.
- Se implementaron modelos SVM, C5.0, NB y RL.
- La evaluación en el conjunto de testing se realizó a partir de accuracy y área bajo la curva ROC.

Tablas de clasificación para modelos RL, NB, SVM y C5.0

Reg Log	+	-
+	**100**	**35**
-	**4**	**8**

AUC.rl= 0.6722
Accu.rl = 0.7346939

Naive Bayes	+	-
+	**88**	**29**
-	**16**	**14**

AUC.nb= 0.5903
Accu.nb = 0.6938776

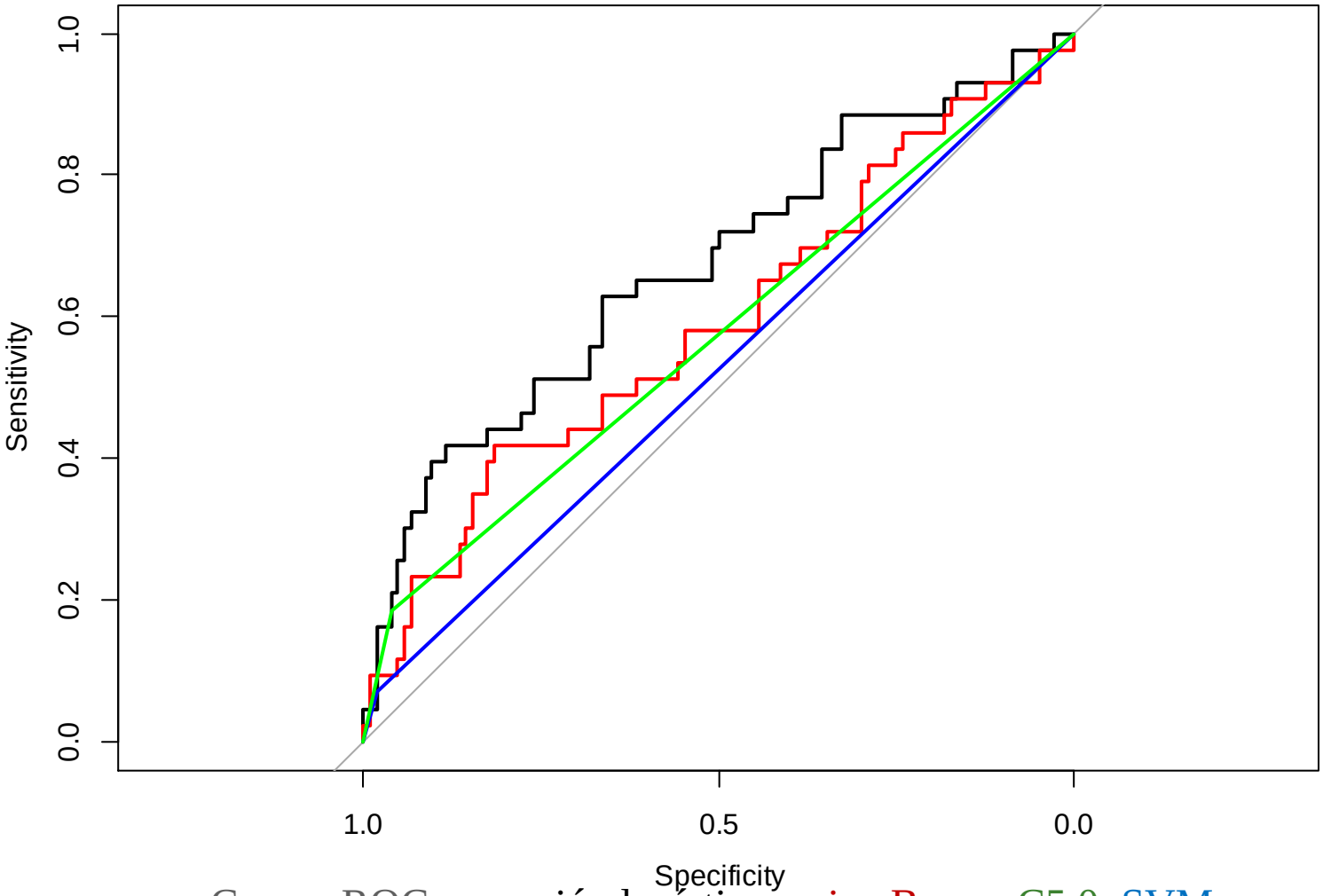
SVM	+	-
+	**97**	**36**
-	**7**	**7**

AUC.svm= 0.5477
Accu.svm = 0.707483

C5.0	+	-
+	**100**	**35**
-	**4**	**8**

AUC.c5= 0.5738
Accu.c5 = 0.7346939

Comparación de modelos con ROC



Curvas ROC: regresión logística; naive Bayes; C5.0; SVM

Resultados

- Los cuatro modelos evaluados son competitivos.
- Los valores de accuracy son similares en los cuatro modelos. Más aún, RL y C5.0 obtuvieron la misma tabla de clasificación!

Pero… es confiable sólo mirar esta tabla?

- Las curvas ROC muestran las diferencias (notar RL vs C5.0). Tomando en cuenta que AUC mide la capacidad predictiva independientemente del punto de corte usado para clasificar, se tiene como…

Modelo ganador:

Regresión Logística (AUC= 0.6722, Accu= 0.73)

Diferencia con Berretta et al: ellos usaron RN con f1:f38 accu=67,6%.

Modelo ajustado por regresión logística

pdrs2C ~ f4 + f10 + f12 + f34 + f38

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-1.82265	0.61762	-2.951	0.003166 **
f4	2.10035	0.61795	3.399	0.000677 ***
f10	0.04963	0.02747	1.807	0.070789 .
f12	-0.06851	0.02979	-2.300	0.021472 *
f34	1.04382	0.70402	1.483	0.138166
f38	-0.69235	0.25071	-2.762	0.005753 **

Obs: las variables más significativas son f4 (relacionada a la frecuencia fundamental) y f38 (potencia espectral altas).

Los signos y coeficientes cuantifican el conocimiento de cómo se relacionan estas variables a la probabilidad de EP:

- *La chance de EP vs noEP se multiplica por 8 por cada unidad de aumento de f4.*
- *La chance de EP vs noEP se reduce a la mitad por cada unidad de aumento de f38.*

Clasificación múltiple en datos Synapse

Se categorizó la variable respuesta del test PDRS según 5 clases.

		Frequency	Percent
Valid	0	180	24,6
	1	141	19,3
	2	151	20,7
	3	130	17,8
	4	129	17,6
	Total	731	100,0

Clase 0: independientes
Clase 1: con baja dificultad de realizar algunos trabajos
Clase 2: con dificultades y lentitud para realizar algunas tareas
Clase 3: alguna dependencia
Clase 4: con discapacidad

Díaz-Pérez. F. García-López. A. Rubio-Sánchez. M and Álvarez-Marquina. A. Using Classification Algorithms for Telemonitoring Parkinson's Disease Severity. in *Advances in Data Mining 17th Industrial Conference on DM*

Clasificación multiclase en datos Synapse

- Se partió el dataset en train/test (fijos para ambos modelos)

```
library(caret)

datos<-read.table(….)

set.seed(2030)

train=sample(seq(length(datos$id)),length(datos$id)*0.80,replace=FALSE)

dataTrain<-datos[train,]

dataTest<-datos[-train,]
```

Clasificación multiclase en datos Synapse

En el conjunto de train se entrenó con *repeated Cross Validation*

```
fitControl <- trainControl(## 10-fold CV
                           method = "repeatedcv",
                           number = 10,
                            repeats = 3)
```

Clasificación multiclase en datos Synapse

Se ajustaron modelos Support Vector Machine y Naive Bayes

```
library(e1071)
nb.Fit = train(pdrs5C~., data = dataTrain, method = "nb",  trControl = fitControl)
nb.pred<-predict(nb.Fit , newdata = dataTest, type = "raw")
svm_cv <- tune("svm", pdrs5C ~ ., data = dataTrain,
               #kernel = "radial",
               kernel = "linear",
              ranges = list(cost = c(0.001,0.01, 0.1, 1, 5, 10, 20), gamma = c(0.5, 1, 5, 10)))
 mejor_modelo <- svm_cv$best.model
```

Se evaluaron modelos en el conjunto de Test:

```
nb.pred<-predict(nb.Fit , newdata = dataTest, type = "raw")
Svm.pred<- predict(object = mejor_modelo, dataTest)
```

Clasificación multiclase

- NB se aplica sin diferencias al caso multiclase.
- SVM se define sólo para 2 clases. En el caso de 5 clases, se ajustan 5*4/2 = 10 SVM binarios (1vs1) y se elige la clase por voto.
- Para SVM es importante recorrer los valores de C que dan un balance sesgo/varianza.
- Al trabajar con *tune* permite elegir el mejor modelo según los hiperparámetros.

Comparación en clasificación multiclase

- Se puede utilizar accuracy con la matriz de confusión. Hay que tener en cuenta que puede ser influenciada por desbalance.
- Puede ser adecuado definir medidas de evaluación costo-sensitivas.

Comparación según tabla de confusión

Real/pred	0	1	2	3	4
0	40	1	0	0	0
1	13	12	0	1	0
2	2	3	23	1	0
3	1	3	2	22	0
4	0	1	0	2	20

Accu.nb =0.7959184

Naive Bayes con repeatedCV, variables f1-f38.

Real/pred	0	1	2	3	4
0	37	4	0	0	0
1	4	20	2	0	0
2	0	0	28	1	0
3	0	0	3	22	3
4	0	0	0	2	21

Accu.svm =0.8707483

SVM lineal con repeatedCV variables f1-f38.

¿es esta una buena forma de compararlos?

Comparación proponiendo costos

Real/pred	0	1	2	3	4
0	40	1	0	0	0
1	13	12	0	1	0
2	2	3	23	1	0
3	1	3	2	22	0
4	0	1	0	2	20

Costo = 62

Naive Bayes con repeatedCV,
todas las variables.

Real/pred	0	1	2	3	4
0	37	4	0	0	0
1	4	20	2	0	0
2	0	0	28	1	0
3	0	0	3	22	3
4	0	0	0	2	21

Costo = 19

SVM lineal con repeatedCV,
todas las variables.

Para la evaluación proponemos:
costo igual al cuadrado de la distancia si la clase real es > a la predicha
costo igual a la distancia si la clase real es < la predicha.

Conclusiones

- La propuesta de análisis de los datos Synapse facilita el seguimiento remoto no invasivo de la progresión de la enfermedad, para lo cual es importante la detección temprana.

- Esto marca la importancia de seleccionar modelos de predicción que tengan en cuenta el tipo de error en la predicción, lo que es posible a partir de propuestas de funciones de costo.

- Las variables que se mostraron relevantes para la construcción del modelo son explícitamente las de la voz, no así sexo, edad, tiempo de EP, entre otras consideradas.